Looking Beyond the Top-1: Transformers Determine Top Tokens in Order

Daria Lioubashevski [1] **Tomer Schlank** [2] **Gabriel Stanovsky** [1] **Ariel Goldstein** [3 4]

Abstract

Uncovering the inner mechanisms of Transformer models offers insights into how they process and represent information. In this work, we analyze the computation performed by Transformers in the layers after the top-1 prediction remains fixed, known as the "saturation event". We expand this concept to top-k tokens, demonstrating that similar saturation events occur across language, vision, and speech models. We find that these events occur *in order* of the corresponding tokens' ranking, i.e., the model first decides on the top ranking token, then the second highest ranking token, and so on. This phenomenon seems intrinsic to the Transformer architecture, occurring across different variants, and even in untrained Transformers. We propose that these events reflect task transitions, where determining each token corresponds to a discrete task. We show that it is possible to predict the current task from hidden layer embedding, and demonstrate that we can cause the model to switch to the next task via intervention. Leveraging our findings, we introduce a token-level early-exit strategy, surpassing existing methods in balancing performance and efficiency and show how to exploit saturation events for better language modeling.

1. Introduction

In recent years, Transformer-based models (Vaswani et al., 2017) have achieved state-of-the-art performance in various tasks across multiple modalities, including text generation, image classification, and automatic speech recognition (Zhang et al., 2023; OpenAI et al., 2023). This has lead to a growing interest in model interpretability, which tries to explain the internal processes that give rise to these remarkable capabilities. In the language domain, investigation into the way the model's predictions are constructed has led to the discovery of *saturation events*, where the model's top-1 prediction is determined in an early layer and remains fixed in subsequent layers (Geva et al., 2022).

In this work, we investigate the following question – *what computation is the Transformer model performing after the saturation event?* Taking inspiration from Frydenlund et al. (2022), we treat the model's output as a ranking over the labels instead of a probability distribution. Using the logit lens (Nostalgebraist, 2020), we project the hidden states of intermediate layers onto the vocabulary space to extract ranking over tokens and analyze the changes over the layers. For the first time, we show that in all tested decoder-only text Transformers (Llama3-8B, GPT2-XL, Mistral-7B, and Falcon-7B) saturation events also take place for the top ranking tokens beyond the top-1 (2nd, 3rd, 4th, etc.).

Surprisingly, we find that they happen *in order* of their ranking, i.e. the second-ranking token is determined only after the first-ranking token, and so forth (see in Figure 1). We then generalize the results across different modalities and Transformer variants, including pretrained Transformers for both vision (encoder-only ViT-L/16) and speech (encoder-decoder Whisper-large). Next, we demonstrate that sequential saturation seems intrinsic to the Transformer architecture, occurring even in an *untrained* randomly initialized model (Llama3-8B).

We propose that this phenomenon is due to a discrete *task-transition* mechanism, where each task i corresponds to the model determining the i-th token in the final ranking, and the transition between one task and the next happens at the corresponding saturation layer. Furthermore, we claim that the task information is encoded in the layer embeddings and that at each saturation layer, a discrete "switch" is flipped, signaling that the relevant token has been determined, causing the model to move on to the next task while keeping this token fixed in subsequent layers. To support this, we show that it is possible to predict the task index from the layer embeddings using a simple logistic regression classifier, and that we can cause the model to transition from the first to

[1]School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem, Israel [2]Department of Mathematics, The University of Chicago, Chicago, US [3]Business School, The Hebrew University of Jerusalem, Jerusalem, Israel [4]Department of Cognitive and Brain Sciences, The Hebrew University of Jerusalem, Jerusalem, Israel. Correspondence to: Daria Lioubashevski <daria.lioubashevski@mail.huji.ac.il>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

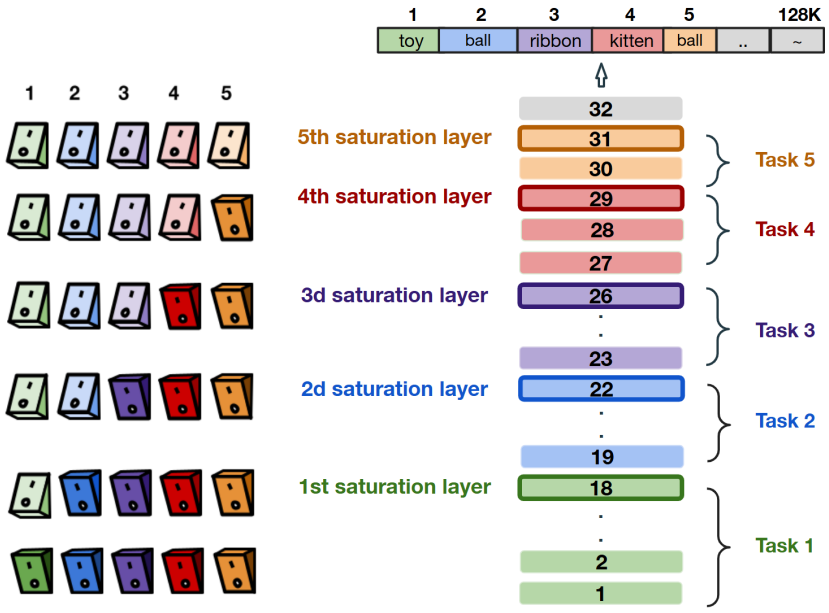


Figure 1. An illustration of the proposed task-transition mechanism wherein the layers of the Transformer perform a changing number of tasks in order, so that task i is determining the i-th token in the final ranking, and the transition between task i and task $i+1$ occurs at the corresponding i-th saturation layer. The transition is akin to a switch being flipped "on" and staying "on" for the remaining layers representing the i-th token being fixed from this point onward.

the second task by "injecting" embeddings from either the top-1 saturation layer or of one of the subsequent layers.

While our primary focus is on uncovering the inner workings of Transformers, we also demonstrate the potential applications of our finding. As a proof of concept, we show that this new understanding of the Transformer lends itself to improving inference efficiency. For this purpose, we introduce a novel token-level early-exit strategy for text generation, where the computation halts after the transition from task 1 to task 2, presuming the top-1 token is finalized. This method outperforms existing early-exit strategies (e.g., softmax-response, Schuster et al., 2022) in balancing computational efficiency and performance. In addition, we show that we can use task information to achieve more accurate language modeling, by demonstrating that in cases where the top-1 prediction is incorrect, the second highest ranking token represents a much more accurate prediction when it reaches saturation than when it does not.

Our main contributions are:

- We find that Transformers tend to decide their top ranking tokens in order, so that the top ranking token is fixed first, then the second-ranking token at a later layer and so on. We show that this occurs across various modalities and variants of the Transformer architecture, and even in untrained randomly initialized models.
- We show that sequential saturation can be explained with a discrete *task-transition* mechanism, encoded in the representation of hidden layers where each task corresponds to determining the next ranking token. We empirically show that it is possible to predict the task index only from internal activations, and that we can cause the model to switch from one task to the next via an intervention procedure.
- We demonstrate that our findings have practical implications by introducing a new token-level early-exit strategy for text generation, as well as insights into better language modeling based on saturation.
- By identifying ordered saturation as a modality-agnostic phenomenon, we take a step toward developing interpretability frameworks that generalize beyond domain-specific observations.

The code for our experiments is available at: https://github.com/daria-lioubashevski/beyond_top1.

2. Experiments

In this section, we first extend the formal definition of top-1 saturation to account for arbitrary i-th ranking token saturation (Section 2.1). Building on this, we formulate two experiments to understand what computation the Transformer performs in the layers after the top-1 saturation event. The

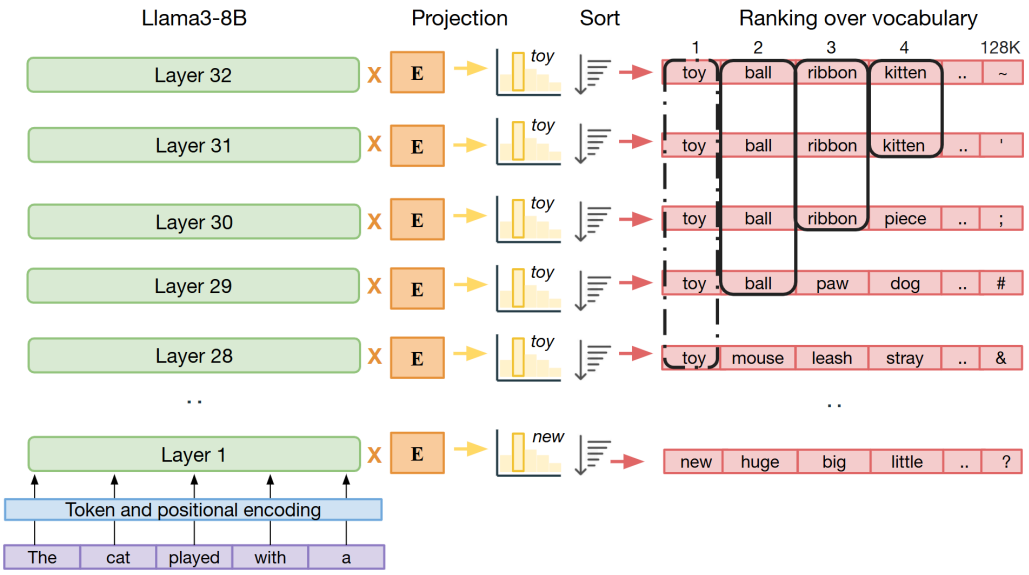


Figure 2. Schematic of our framework and visualization of the ordered saturation of the top-k tokens on Llama3-8B. The hidden states from each layer are projected onto the vocabulary space using the unembedding matrix E, then sorted in descending order and treated as rankings. The saturation effect is marked separately for each token in the top-4 of the final ranking, emphasizing the fact that the 2nd token saturates *after* the 1st token, the 3d token saturates *after* the 2nd token and so on. The dashed line represents the previously established saturation event of the top-1 token.

first experiment (Section 2.2) leverages our definition to develop a metric capturing the extent to which top tokens are saturated in order. The second experiment (Section 2.3) uses a probing approach to test whether it is possible to determine the rank of the token currently being determined by the model solely from intermediate layer activations.

2.1. Defining Saturation Layers

Definition 2.1 (1st Saturation Layer; Geva et al., 2022)**.** The saturation event occurs at layer l (from here on referred to as the "1st saturation layer") for index i in the input if the top-1 prediction of the model remains constant in all subsequent layers after l. Formally, given a model with N layers, a saturation event occurs at layer $l \leq N-1$ if for all layers l' s.t. $l < l' \leq N$ the top-1 token in the ranking induced by that layer remains unchanged. For example, the saturation event marked with a dashed line in Figure 2 occurs on layer 28, since in subsequent layers the top predicted token ("toy") remains constant.

Definition 2.2 (k-th Saturation Layer)**.** Here, we are interested in examining model behavior beyond the determination of the top ranking token and so naturally extend the definition of top-1 Saturation (Definition 2.1) to capture the layer at which the k-th top token is determined and remains fixed. Formally, the saturation event for the k-th top token at index i in the input occurs at layer $l^k \leq N-1$ (from here on referred to as the " k-th saturation layer") if for all following layers l' s.t. $l^k < l' \leq N$ the token in position k in the ranking induced by that layer remains unchanged. We note that the saturation event defined in (Geva et al., 2022) happens at l^1. For example, in Figure 2, $l^1 = 28$ as that is where the top token ("toy") is determined; $l^2 = 28$, since the second-most probable token ("ball") is determined at layer 29; and similarly $l^3 = 30$, since the third-most probable token ("ribbon") is determined at layer 30, etc.

2.2. Examining the Order of Saturation Layers

We investigate whether the saturation layers of the top-k tokens $l^1, l^2, \ldots l^k$ are arranged in order, i.e., whether the saturation of the first token happens before the saturation of the second token, the saturation of the second token happens before the saturation of the third token and so on[1]. To this end, for each token in the input we first calculate these saturation layers for $k = 1, \ldots, 5$ according to Definition 2.2 and then for each k compute the rank of the saturation layer of the k-th top token. We use $k = 5$ to ensure consistency across models, as it is the highest value of k where the k-th token reaches saturation in at least 5% of input tokens for all the models analyzed. Following our example from Figure 2, we have $l^1 = 28, l^2 = 29, l^3 = 30, l^4 = 31$,[2]

[1] In all of our experiments, we only consider tokens in the input where the 1st saturation layer satisfies that $l^1 \leq 0.85 \cdot N$, to ensure that there are enough layers after it for meaningful analysis.

[2] l^5 is ill-defined in this case as the 5-th token doesn't reach saturation before the last layer.

and their ranking is $[1, 2, 3, 4]$, since $l^1 < l^2 < l^3 < l^4$. If the tokens reach saturation in order of ranking, as they do in this case, we would expect the average rank of the saturation layers to increase monotonically with k. To statistically validate this phenomenon we use a stricter version of Kendall's τ coefficient that treats ties as disagreements, allowing us to quantify the alignment between the token ranking and the ranking of their corresponding saturation layers (see Appendix A.3).

2.3. Probing for Task Transition

We argue that the mechanism underlying the saturation of the top-k tokens in order is one of task transition, such that determining the identity of each token in the final ranking is a separate task, and the model performs them sequentially: first determining the identity of 1st token, then the identity of the 2nd token, and so on, and that the transition from one task to the next occurs at the corresponding saturation layer. Importantly, we do not claim that the model does not process token i before task i, rather it may begin accumulating evidence for token i earlier, but the token is fixed at the end of the task at the corresponding saturation layer. Additionally, we claim that the specific task number can be inferred from the model embedding at each layer, and that this information is independent of the context or the specific token predicted by the model.

To test our hypothesis, we perform a type of probing in which we train a simple one-versus-all multi-class logistic regression classifier to predict the number of the task the model is "working on" from the hidden layer embeddings. We collect the data for training by extracting the model's hidden states during inference and categorize them into 5 classes according to the saturation layers of the top-5 tokens for each instance. This means that for a given input, embeddings from layers up to (and including) the 1st saturation layer are classified as belonging to task 1, embeddings from layers from the next layer until the 2nd saturation layer are classified as belonging to task 2, and so forth.

For example, in the case of the token "a" as depicted in Figure 2, the embeddings from layers 1 through 28 would be classified as belonging to task 1; the embedding of layer 29 would be classified as belonging to task 2; the embedding of layer 30 as belonging to task 3; the embedding of layer 31 as belonging to task 4, and as the model reaches the last layer directly afterward there would be no embedding classified as belonging to task 5.

We balance the training data to ensure equal representation of embeddings from all layers in each class. To confirm that task encoding arises from the model's embeddings rather than the classifier's weights, we create a control setting by generating random vectors for each class, matching the dimensions, mean, and variance of the corresponding layer embeddings.[3]

3. Results

To show the robustness of our findings, we test pretrained Transformer models on corresponding datasets from three modalities: text, vision and speech.

Top-1 saturation events are common. Saturation events are quite common, with 10% to 60% of all input tokens reaching top-1 saturation, according to our defintion, in all models tested (see Appendix A.2 for more statistics).

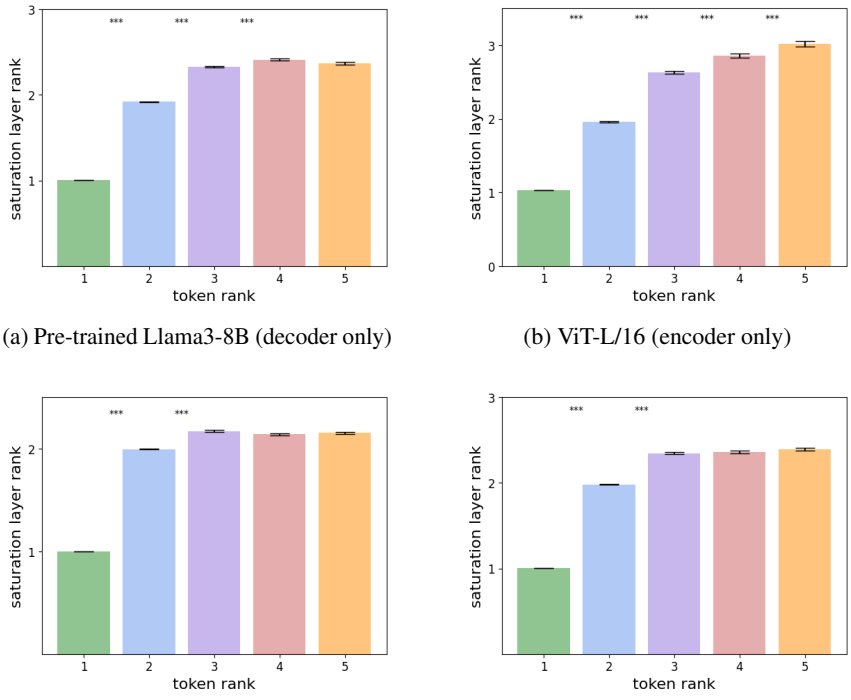
3.1. Text Transformer

We conduct our experiments to examine the order of saturation in a wide range of LLMs: GPT2-XL (Radford et al., 2019), 8-bit quantized versions of Llama3-8B (Wendler et al., 2024), Mistral-7B (Jiang et al., 2023) and Falcon-7B (Almazrouei et al., 2023). We employ a set prompt to present each task's questions, answer choices, and correct answer, ensuring a uniform input structure (see Appendix A.1 for prompt details).

Tokens reach saturation in order of ranking. We use 1K randomly sampled questions from MMLU test split which represent 60K-100K tokens (depending on the model). In all tested models, we find that tokens tend to reach saturation in order. As an example, Figure 3a shows the average rank of the k-th saturation layer for each k for the Llama3-8B model. This value increases monotonically with k, and the difference between each two consecutive token ranks is statistically significant with $p < 0.001$ based on a pairwise independent samples t-test. All other models show a similar trend, see Appendix A.4.

Task number can be predicted from model embeddings. After extracting embeddings from 500 randomly sampled questions we split the data into train and test using 5-fold cross validation, and report the mean and standard error of the accuracy. Table 1 shows that the logistic regression classifier trained on embeddings extracted from pre-trained Llama3-8B model achieves very high accuracy, while the classifier trained on the random embeddings in the control setting performs approximately at chance level (see Appendix A.7 for accuracy and ROC-AUC scores per class). From this we conclude that the representations of the hidden layers across examples encode task specific information and that the saturation layers as we defined them are the points of transition between those tasks. The same qualitative results are observed for all other models, see Appendix A.5.

[3]The number of tasks per model is set to the maximum where data balancing yields at least 10 embeddings per class from at least 4 different layers.



(a) Pre-trained Llama3-8B (decoder only)

(b) ViT-L/16 (encoder only)

(c) Whisper-large (full-Transformer)

(d) Randomly initialized Llama3-8B

Figure 3. Average rank of the k-th saturation layer among the saturation layers for k=1,..,5 with standard error bars. Asterisks indicate statistically significant differences between consecutive token ranks (*** represents $p < 0.001$), based on an independent samples t-test.

3.2. Vision Transformer

Encoder-only image-classification ViTs (Dosovitskiy et al., 2021) take as input a sequence of linear projections of equal-sized image patches with added position embedding and a special "class token" denoted [CLS]. Following the work of (Vilas et al., 2023) we use a version of the logit lens adapted to ViT to project the hidden state representations of each layer in the encoder onto the class embedding space using the output embedding matrix. Importantly this is done only for the [CLS] token for each image under the assumption that it best represents the model's prediction, since during ViT's pretraining the only token projected onto the class-embedding space is the [CLS] token from the last layer.

For our experiments we use the ViT-L/16 variant pretrained on ImageNet-21k and fine-tuned on ImageNet 2012, and run inference on 5K randomly sampled images from the CIFAR-10 (Krizhevsky et al., 2009) dataset. Figure 3b demonstrates the high correspondence between saturation layer and token rank, supporting our claim that in this domain as well as in text the saturation layers are highly ordered. Furthermore, Table 1 shows that the task index can be predicted from the hidden layer activations with high accuracy well above chance and control setting. We further validate our findings in a multi-modal setting using the vision-language model LLaVa (Liu et al., 2023) on the MMMU (Yue et al., 2024) visual question answering benchmark, see Appendix A.8.

3.3. Speech Transformer

Whisper (Radford et al., 2023) is an encoder-decoder Transformer model trained on many different speech processing tasks, including ASR. Although recently there have been attempts to increase efficiency in ASR, such as Malard et al. (2023), the concept of early exit has yet to be explored in this setting, and to the best of our knowledge there has not been work done concerning saturation events in speech models. We adapt the logit lens and apply it *only* to the decoder stack of Whisper-large, under the assumption that representations in the encoder stack are inherently different and projecting them onto the token vocabulary space would not be meaningful. For our dataset we randomly sample 5K audios from LibriSpeech (Panayotov et al., 2015).

In addition to reproducing for the first time in speech the classical top-1 saturation event established in language and vision models in previous work, we also show in Figure 3c evidence for the tendency of the top-k tokens to reach saturation in order in this model as well, albeit only up to the third token. We suspect that the deterioration in order for later tokens arises from the fact that all decoder layers are conditioned via cross-attention on the final layer of the encoder. This global conditioning may interfere with the task transition mechanism by "blurring the lines" between tasks, effectively entangling task-specific representations. Supporting this hypothesis, in Appendix A.9 we replicate

Table 1. Accuracy of task number logistic regression classifier showing that in all modalities the layer embeddings contain information about the task number. Asterisks indicate statistically significant accuracy (* represents $p < 0.001$), based on an Binomial Distribution probability test. P stands for pre-trained, R stands for randomly initialized.

Model	Layer Emb.	Random Emb.	Chance Level
LLaMA3-8B (P)	**88.1**$^*\pm$ 0.4	33.1 $\pm$ 0.5	33.3
LLaMA3-8B (R)	**79.2**$^*\pm$ 0.3	34.1 $\pm$ 0.4	33.3
ViT-L/16	**63.8**$^*\pm$ 0.1	21.0 $\pm$ 0.5	20.0
Whisper-large	**52.7**$^*\pm$ 0.1	24.5 $\pm$ 0.4	25.0

our findings using Qwen-Audio (Chu et al., 2023), an audio model in which the output of the audio encoder is used only as input to the first decoder layer, and observe that the ordered saturation persists up to the 4th token. Even so, Table 1 shows that the task index can be predicted from Whisper's decoder layers' embeddings with accuracy much higher than chance or that achieved in the control setting.

4. Analysis

We have shown that top-k tokens tend to reach saturation in order of their ranking, as well as the plausibility of the underlying task transition mechanism over multiple modalities and variants of Transformers: decoder only, encoder only and full Transformer; in section 4.1 we argue that this phenomenon is inherent to the architecture itself and in section 4.2 we delve deeper into the way the model transitions between tasks, demonstrating that we can cause the model to switch to the next task using an intervention procedure.

4.1. Untrained Transformers Also Determine in Order

We repeat our experiments on an untrained Llama3-8B with randomly initialized weights on the same amount of randomly sampled questions from MMLU dataset as with the pretrained model. Surprisingly, Figure 3d shows that the top-k tokens tend to reach saturation in order up to the 3th token. In addition, Table 1 shows that the task transition classifier's accuracy is more than $2x$ times higher than chance or that of the control setting. The ability of the classifier to extract the task index from the hidden layers' representations in this setting is especially remarkable, demonstrating that despite the randomness of the weights as well of the identities of the predicted tokens, there is still highly ordered information encoded in the model originating only from the constraints of the architecture.

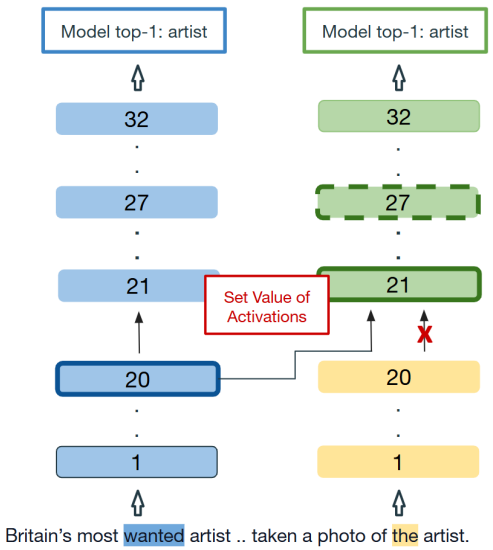


Figure 4. By injecting the output from the top-1 saturation layer of "the" as input to the subsequent layer of "artist", we trigger a saturation at the injected layer (21) in the post-intervention run, without altering the top-1 prediction. Saturation layers are marked in bold, saturation layer in the original run for "the" token marked with dashes. The use of activations from adjacent layers is not depicted for the sake of clarity.

4.2. Injecting Saturation Layer Activations Causes Task Switch

Using the probing analysis, we demonstrated that the tasks, as we defined them, are distinct enough to be separated by a simple classifier, that saturation layers mark the boundaries between them, and that the task index is encoded in the hidden layer embeddings. We argue that in addition, each saturation layer encodes the signal to transition to the next task, and all subsequent layers contain the information that the previous task has been completed and that the relevant token is fixed. This can be thought of as switch being flipped "on" for each token that reaches saturation, and remaining "on" from the saturation layer onwards.

To causally validate this claim, taking inspiration from Stolfo et al. (2023), we perform an intervention (visualized in Figure 4) in which we "inject" the output from the 1st saturation layer of sample s_1 as input into the subsequent layer in the run on sample s_2 and check how this affects the 1st saturation layer of s_2. If these activations contain the signal to switch to the next task, we expect this intervention to cause the model to switch to task 2 at the injected layer in the post-intervention run, i.e. in the new post-intervention run the 1st saturation layer should be the one on which the intervention is performed which is $l^1(s_1) + 1$. To minimize the effect of confounding factors, we choose pairs of samples s_1 and s_2 that share context and where the original

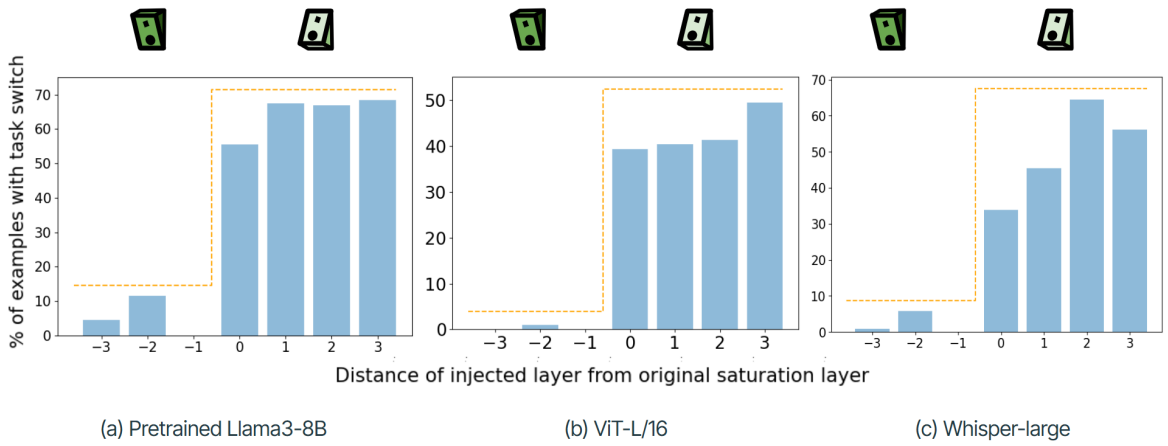


Figure 5. Flipping the Top-1 Switch. The percentage of examples where the top-1 saturation occurred at the injected layer after the intervention, shown as a function of the layer from which the injected activations were taken, relative to the original saturation layer (e.g., -2 means activations were taken from two layers before the original saturation layer).

top-1 prediction of the model is the same, but there is a big difference in their 1st saturation layers s.t. $l^1(s_1) < l^1(s_2)$.

In the example depicted in Figure 4 s_1 = "*wanted*" and s_2 = "*the*", and for both the model's top-1 next-word prediction is "artist", but $l^1(s_1) = 20$ while $l^1(s_2) = 27$. Injecting the output of layer $l^1(s_1)$ into the subsequent layer (21) in the run for s_2 should cause the model to switch to task 2, resulting in layer 21 being the new 1st saturation layer post intervention.

Moreover, we would expect the same thing to happen when injecting activations from a layer l *after* the 1st saturation layer, i.e. $l > l^1(s_1)$, since they should contain the information that the top-1 token is fixed. On the other hand, activations from a layer l' *before* the 1st saturation layer, i.e. $l' < l^1(s_1)$ should not result in saturation at the injected layer as the switch is still "off" in our analogy, indicating to the model that it still working on task 1. To test this, we repeat the same steps with activations from 3 layers before and after the 1st saturation layer $[l^1(s_1) - 3, l^1(s_1) + 3]$ each time injecting them as input into the subsequent layer.

Figure 5(a) shows the results of this procedure performed using pre-trained Llama3-8B on 200 token pairs taken from 10 randomly sampled texts from the CNN/DM dataset[4] (Hermann et al., 2015), Figure 5(b) shows similar results reproduced using ViT-L/16 on 200 pairs of images from CIFAR-10 dataset, and Figure 5(c) shows the results of the intervention on Whisper-large on 200 token pairs from 100 randomly sampled audios from LibriSpeech[5]. There is a stark difference in the effect the injected activations have on the 1st saturation layer post-intervention when the activations are taken from the 1st saturation layer in the original run or one of the following layers, compared to the layers before it. When the injected activations are taken from an earlier layer, the new top-1 saturation almost never occurs at the injected layer, whereas when the injected activations are taken from the saturation layer or a later layer the top-1 saturation occurs at the injected layer in a high percentage of cases. This drastic change resembles a step function, and is in line with our description of a switch being flipped "on" at the 1st saturation layer and remaining on in all subsequent layers, indicating to the model to switch to the next task and keep the top-1 constant. These results are consistent across a range of LLMs, as shown in Appendix A.11.

5. Practical Implications

While the primary focus of this work is on understanding the mechanisms underlying saturation events, we demonstrate how these insights can inform practical strategies for improving computational efficiency and language modeling.

5.1. New Early-Exit Strategy

We propose a new token-level dynamic inference method based on the task-transition classifier described in Section 2.3, where the early exit layer for each token is defined as the earliest layer which is predicted to belong to task 2 by the classifier. The idea being that once the model has transitioned into the second task, it has finished with the first task of determining the top-1 token. To demonstrate the viability of this method, we compare it to two other local confidence measures introduced by Schuster et al. (2022): softmax response (the difference between the top two values

[4]We use texts from CNN/DM and not MMLU for this experiment as they tend to be longer and have more pairs that fit our criteria for intervention

[5]See Appendix A.10 for a formal description of the procedure as well as adaptations for vision and speech modalities

of the logits after softmax) and hidden-state saturation (cosine similarity between two consecutive layer embeddings), both recently found to be competitive with other early exiting methods (Zhou et al., 2024). Since dynamic decoding is not the focus of this paper, we calculate the metrics for each measure while propagating states from the layers after the "early exit" as in regular inference.[6]

Table 2 shows our results on a pre-trained Llama3-8B model and 100 randomly sampled texts from CNN/DM dataset, showcasing the ability of our probing classifier to generalize across datasets as it was trained on texts from MMLU benchmark (see Section 3.1). We evaluate the model on next-word prediction, and compute the speedup ratio for each method as the total number of layers in the model divided by the number of layers it uses for each token.[7] For the two other confidence measures we calculate these metrics at various thresholds (see details in Appendix A.12), while in our measure the class is selected based on the highest predicted score among all classes. Our strategy outperforms the other two when considering the trade-off between next-word prediction accuracy and speedup ratio, and requires no training besides that of a simple logistic regression classifier on a relatively small amount of data. We find the difference in speedup between our strategy and the other methods to be statistically significant ($p < 0.001$) using an independent samples t-test.

5.2. Improved Language Modeling

Popular decoding methods in language generation such as top-k (Fan et al., 2018) or top-p (Holtzman et al., 2020b) sample the next token from the top ranking tokens. Based on our task-transition mechanism and the assumption that the tasks represent relevant computation, we argue that top ranking tokens that are determined in the last layer represent less meaningful predictions, since the model only had enough layers for the first task in these instances. To test this hypothesis, we compare the accuracy of the second, third, and fourth ranked tokens in the next-word prediction task across two conditions: (1) the token in question reaches saturation i layers before final layer (with $2 \leq i \leq 6$); (2) the token does not saturate, and is determined only in the last layer. In all cases, we restrict the analysis to examples where the top-1 prediction is incorrect.

Using 100 randomly sampled texts from CNN/DM dataset and pre-trained Llama-8B predictions, we show in table 3 that in the first condition the accuracy is much higher than in the second condition. A two proportion z-test indicates a statistically significant difference between the groups ($p < 0.001$) even when correcting for multiple comparisons for $2 \leq i \leq 6$ in the 2nd and 3rd tokens, and for and $2 \leq i \leq 5$ in the 4th token. This supports our claim that top-k tokens that reach true saturation are more plausible predictions than those that are determined in the last layer, which has potential implications for decoding strategies which consider tokens beyond the top-1.

[6]This is an informative comparison between the measures, as the effect of a state copying mechanism for skipped layers on model's performance is negligible (Schuster et al., 2022).

[7]We apply all strategies starting from layer 15, since our classifier was trained only on the intermediate layers of the network.

Table 2. Highest accuracy and corresponding speedup-ratio achieved by each early-exit strategy. Asterisks indicate statistically significant accuracy (* represents $p < 0.001$), based on an independent samples t-test.

	Softmax Response	State Saturation	Ours
Accuracy	40.6± 0.6	24.3± 0.4	**43.3** ± 0.7
Speedup Ratio	1.07± 0.002	1.001± 0.000	**1.11*** ±0.009

6. Related Work

There are many ways of thinking about the role of intermediate layers in Deep Neural Networks (DNNs) in general, and Transformers in particular. The iterative inference hypothesis interprets each layer as an iteration from an iterative and convergent process (Simoulin & Crabbé, 2021), suggesting that each layer incrementally refines the hidden representation by gradually shaping the next token prediction (Geva et al., 2022; Belrose et al., 2023; Rushing & Nanda, 2024). We argue that our findings challenge this view, given the discrete nature of the tasks in the proposed task-transition mechanism and the sharp transitions between them.

Pruning is another approach, focused on mitigating the redundancy inherent to large machine learning models by removing unnecessary parameters. Recent work has applied structured pruning methods to Transformer based LLMs, dropping whole modules, from self-attention layers (Ben-Artzy & Schwartz, 2024; He et al., 2024) to full Transformer blocks (Sun et al., 2024; Men et al., 2024). These studies often focus on the middle layers of the model, and claim to reduce memory and computation costs without degrading performance. However, these works evaluate the accuracy before and after pruning based only on the top-1 prediction of the model, even though stochastic generation methods such as top-p (Holtzman et al., 2020a) and top-k (Fan et al., 2018) are preferable to deterministic decoding in certain settings such as open-ended tasks, as they produce more coherent and varied text (Shi et al., 2024). In light of this and of our results regarding the sequential saturation of top ranking tokens, we suggest that future work takes this into account, since what may seem as redundancy is actually necessary computation not reflected in the measured metric.

Table 3. LLaMA3-8B: Accuracy (percentage) of next-word prediction for the 2nd, 3rd, and 4th ranked tokens when the top-1 token is incorrect
(comparing no saturation to saturation i layers before output).

Token rank	No saturation	Saturation i layers before output				
		$i=2$	$i=3$	$i=4$	$i=5$	$i=6$
$2nd$	25.2	38.1	48.6	37.6	37.6	33.0
$3d$	17.0	23.3	26.9	28.9	29.2	31.0
$4th$	13.4	16.8	19.7	20.5	19.5	18.3

The logit lens has also been used to study intermediate layers in a wide variety of interpretability papers (Yang et al., 2024; Wendler et al., 2024; Halawi et al., 2024). Despite this, (Belrose et al., 2023) claim that it can produce implausible results due to the difference in representations between layers. To address this issue they introduce the "tuned lens", in which an affine transformation is learned for each layer in the model with a distillation loss, so that its image under the unembedding matrix matches the final layer logits as closely as possible. Although this method may be better at approximating final top-1 prediction from intermediate layers, our work highlights why this might actually be a disadvantage when attempting to gain insights into the computational process of the Transformer, as it could obscure the changing dynamics of the lower ranked tokens.

7. Discussion

This work contributes to a deeper understanding of how Transformer models construct predictions over layers. We find that the top-k tokens (for $k > 1$) go through saturation events in the order of their ranking. This phenomenon is inherent to the Transformer architecture, occurring in untrained models and is robust over multiple modalities. We provide evidence in support of task transition as the underlying mechanism, showing that we can predict task index from the hidden layers' embeddings, as well as cause the model to switch from the first task to the second via an intervention procedure. Our findings also show promise in improving LLMs' efficiency and performance as suggested by the preliminary results in Section 5.

Our findings also carry broader implications for how we understand the internal structure of Transformer models. A common line of work in interpretability of LLMs aims to assign distinct computational roles to different layers, often proposing a hierarchical progression of feature complexity (Tenney et al., 2019; van Aken et al., 2019; Geva et al., 2021; 2022), suggesting that earlier layers specialize in shallow or syntactic features, while later layers capture more semantic or task-specific information. However, our probing results indicate that the relationship between layer depth and computation is more dynamic: the same layer can participate in qualitatively different mechanisms depending on the structure of the task, challenging the view of fixed layer specialization.

In addition, the emergence of ordered saturation in an untrained Transformer raises fundamental questions about the architectural biases baked into the model. We observe that this behavior not only appears prior to training (potentially influenced by weight initialization) but also persists after pre-training, suggesting a constraint on the model's computation. This opens the door to future work examining how such structural biases influence representational dynamics and probability distributions in language models.

Limitations While our analysis sheds light on the high-level mechanism behind the ordered saturation of top-k tokens, several questions remain open. First, the precise architectural components responsible for this phenomenon are still unclear. Targeted ablation studies could help localize the origin of the effect within the Transformer architecture. Second, a more concrete explanation is needed for how the model keeps the "chosen" tokens constant in the layers after the corresponding saturation event. Third, we did not consider whether the model had seen the experimental data during training and how this might affect ordered saturation. Furthermore, since we focused solely on Transformer architectures, it remains to be explored whether other DNN types also determine their top-ranking tokens in order. Recurrent Neural Networks might be of particular interest due to their mathematical equivalence to decoder only Transformers (Oren et al., 2024), and based on previous work successfully applying the logit lens to them to extract meaningful predictions from intermediate layers (Paulo et al., 2024).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgments

We would like to thank Nir Rosenfeld, Omri Abend and Mariano Schain for the invaluable insights, as well as Noam Dahan and Timna Wharton Kleinman for their helpful feedback.

References

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., Goffinet, É., Hesslow, D., Launay, J., Malartic, Q., et al. The falcon series of open

language models. *ArXiv preprint*, abs/2311.16867, 2023. URL https://arxiv.org/abs/2311.16867.

Belrose, N., Furman, Z., Smith, L., Halawi, D., Ostrovsky, I., McKinney, L., Biderman, S., and Steinhardt, J. Eliciting latent predictions from transformers with the tuned lens. *ArXiv preprint*, abs/2303.08112, 2023. URL https://arxiv.org/abs/2303.08112.

Ben-Artzy, A. and Schwartz, R. Attend First, Consolidate Later: On the Importance of Attention in Different LLM Layers. *arXiv*, 2024. doi: 10.48550/arXiv.2409.03621.

Chu, Y., Xu, J., Zhou, X., Yang, Q., Zhang, S., Yan, Z., Zhou, C., and Zhou, J. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *ArXiv preprint*, abs/2311.07919, 2023. URL https://arxiv.org/abs/2311.07919.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Fan, A., Lewis, M., and Dauphin, Y. Hierarchical neural story generation. In Gurevych, I. and Miyao, Y. (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, Melbourne, Australia, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082. URL https://aclanthology.org/P18-1082.

Frydenlund, A., Singh, G., and Rudzicz, F. Language modelling via learning to rank. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 10636–10644. AAAI Press, 2022. URL https://ojs.aaai.org/index.php/AAAI/article/view/21308.

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, Online and Punta Cana, Dominican Republic, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL https://aclanthology.org/2021.emnlp-main.446.

Geva, M., Caciularu, A., Wang, K., and Goldberg, Y. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In Goldberg, Y., Kozareva, Z., and Zhang, Y. (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 30–45, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.3. URL https://aclanthology.org/2022.emnlp-main.3.

Halawi, D., Denain, J., and Steinhardt, J. Overthinking the truth: Understanding how language models process false demonstrations. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Tigr1kMDZy.

He, S., Sun, G., Shen, Z., and Li, A. What matters in transformers? not all attention is needed. *ArXiv preprint*, abs/2406.15786, 2024. URL https://arxiv.org/abs/2406.15786.

Hermann, K. M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. Teaching machines to read and comprehend. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 1693–1701, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/afdec7005cc9f14302cd0474fd0f3c96-Abstract.html.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020a. URL https://openreview.net/forum?id=rygGQyrFvH.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020b. URL https://openreview.net/forum?id=rygGQyrFvH.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *ArXiv preprint*,

abs/2310.06825, 2023. URL https://arxiv.org/abs/2310.06825.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html.

Malard, H., Zaiem, S., and Algayres, R. Big model only for hard audios: Sample dependent whisper model selection for efficient inferences. *ArXiv preprint*, abs/2309.12712, 2023. URL https://arxiv.org/abs/2309.12712.

Men, X., Xu, M., Zhang, Q., Wang, B., Lin, H., Lu, Y., Han, X., and Chen, W. Shortgpt: Layers in large language models are more redundant than you expect. *ArXiv preprint*, abs/2403.03853, 2024. URL https://arxiv.org/abs/2403.03853.

Nostalgebraist. Interpreting gpt: the logit lens, 2020.

OpenAI, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., and et al. Gpt-4 technical report, 2023. URL https://arxiv.org/abs/2303.08774.

Oren, M., Hassid, M., Adi, Y., and Schwartz, R. Transformers are multi-state rnns. *ArXiv preprint*, abs/2401.06104, 2024. URL https://arxiv.org/abs/2401.06104.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pp. 5206–5210. IEEE, 2015. doi: 10.1109/ICASSP.2015.7178964. URL https://doi.org/10.1109/ICASSP.2015.7178964.

Paulo, G., Marshall, T., and Belrose, N. Does transformer interpretability transfer to rnns? *ArXiv preprint*, abs/2404.05971, 2024. URL https://arxiv.org/abs/2404.05971.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. Robust speech recognition via large-scale weak supervision. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518. PMLR, 2023. URL https://proceedings.mlr.press/v202/radford23a.html.

Rushing, C. and Nanda, N. Explorations of self-repair in language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=5ZwEifshyo.

Schuster, T., Fisch, A., Gupta, J., Dehghani, M., Bahri, D., Tran, V., Tay, Y., and Metzler, D. Confident adaptive language modeling. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/6fac9e316a4ae75ea244ddcef1982c71-Abstract-Conference.html.

Shi, C., Yang, H., Cai, D., Zhang, Z., Wang, Y., Yang, Y., and Lam, W. A thorough examination of decoding methods in the era of llms. *ArXiv preprint*, abs/2402.06925, 2024. URL https://arxiv.org/abs/2402.06925.

Simoulin, A. and Crabbé, B. How many layers and why? An analysis of the model depth in transformers. In Kabbara, J., Lin, H., Paullada, A., and Vamvas, J. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pp. 221–228, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-srw.23. URL https://aclanthology.org/2021.acl-srw.23.

Stolfo, A., Belinkov, Y., and Sachan, M. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7035–7052, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.435. URL https://aclanthology.org/2023.emnlp-main.435.

Sun, Q., Pickett, M., Nain, A. K., and Jones, L. Transformer layers as painters. *ArXiv preprint*, abs/2407.09298, 2024. URL https://arxiv.org/abs/2407.09298.

Tenney, I., Das, D., and Pavlick, E. BERT rediscovers the classical NLP pipeline. In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452.

van Aken, B., Winter, B., Löser, A., and Gers, F. A. How does BERT answer questions?: A layer-wise analysis of transformer representations. In Zhu, W., Tao, D., Cheng, X., Cui, P., Rundensteiner, E. A., Carmel, D., He, Q., and Yu, J. X. (eds.), *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, pp. 1823–1832. ACM, 2019. doi: 10.1145/3357384.3358028. URL https://doi.org/10.1145/3357384.3358028.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Vilas, M. G., Schaumlöffel, T., and Roig, G. Analyzing vision transformers for image classification in class embedding space. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/7dd309df03d37643b96f5048b44da798-Abstract-Conference.html.

Wendler, C., Veselovsky, V., Monea, G., and West, R. Do llamas work in english? on the latent language of multilingual transformers. *ArXiv preprint*, abs/2402.10588, 2024. URL https://arxiv.org/abs/2402.10588.

Yang, S., Gribovskaya, E., Kassner, N., Geva, M., and Riedel, S. Do large language models latently perform multi-hop reasoning? *ArXiv preprint*, abs/2402.16837, 2024. URL https://arxiv.org/abs/2402.16837.

Yue, X., Ni, Y., Zheng, T., Zhang, K., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., Wei, C., Yu, B., Yuan, R., Sun, R., Yin, M., Zheng, B., Yang, Z., Liu, Y., Huang, W., Sun, H., Su, Y., and Chen, W. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 9556–9567. IEEE, 2024. doi: 10.1109/CVPR52733.2024.00913. URL https://doi.org/10.1109/CVPR52733.2024.00913.

Zhang, Y., Han, W., Qin, J., Wang, Y., Bapna, A., Chen, Z., Chen, N., Li, B., Axelrod, V., Wang, G., et al. Google usm: Scaling automatic speech recognition beyond 100 languages. *ArXiv preprint*, abs/2303.01037, 2023. URL https://arxiv.org/abs/2303.01037.

Zhou, Z., Ning, X., Hong, K., Fu, T., Xu, J., Li, S., Lou, Y., Wang, L., Yuan, Z., Li, X., et al. A survey on efficient inference for large language models. *ArXiv preprint*, abs/2404.14294, 2024. URL https://arxiv.org/abs/2404.14294.

A. Appendix

A.1. Prompt Format

When using questions from MMLU dataset in our experiments we employ the following prompt to present each task's questions, answer choices, and correct answer, ensuring a uniform input structure.

Prompt format:

Question: <QUESTION>
A. <CHOICE A>
B. <CHOICE B>
C. <CHOICE C>
D. <CHOICE D>
Answer: <ANSWER>

A.2. Saturation statistics

Table 4 shows the percent of input tokens which reach top-1 saturation in the first 85% layers of the model across all models we tested. For all text-Transformers we use 1K randomly sampled questions from MMLU dataset with the prompt format described in A.1, for ViT we use 1K random images from CIFAR-10 dataset, and for Whisper we use 1K random audios from LibriSpeech dataset.

For Llama3-8B, for example, we find that 31.2% of all input tokens reach top-1 saturation as we defined it. Additionally, we show in Figure 6 that the tokens that reach top-1 saturation belong to all different parts of speech (POS) including content words, with over 30% of them being nouns.

A.3. Stricter Kendall's tau

We define a version of Kendall's tau coefficient measuring the ordinal association between two tanking, where one-sided ties are considered discordant unlike the regular metric, where ties are typically either ignored or handled as neutral, meaning they neither count as concordant nor discordant. This is done to discount cases where two or more tokens reach saturation at the same layer. The coefficient takes values in the range $[-1, 1]$ where values close to 1 indicate strong agreement, and values close to -1 indicate strong disagreement between the rankings.

Formally, given two rankings $x = (x_1, x_2, \ldots, x_n)$ and $y = (y_1, y_2, \ldots, y_n)$, let pair$(i, j)$ be a pair of indices where $1 \leq i < j \leq n$.

We define the pair as *concordant* if the rankings in both sequences agree, meaning:

$$\begin{aligned} &(x_i > x_j \text{ and } y_i > y_j) \quad \text{or} \\ &(x_i < x_j \text{ and } y_i < y_j) \quad \text{or} \\ &(x_i = x_j \text{ and } y_i = y_j) \end{aligned}$$

Table 4. Percent of input tokens that reach top-1 saturation.

Model	% tokens
Llama3-8B	31.24
Mistral-7B	43.36
Falcon-7B	11.75
GPT2-XL	58.42
ViT-L/16	44.70
Whisper-large	10.92

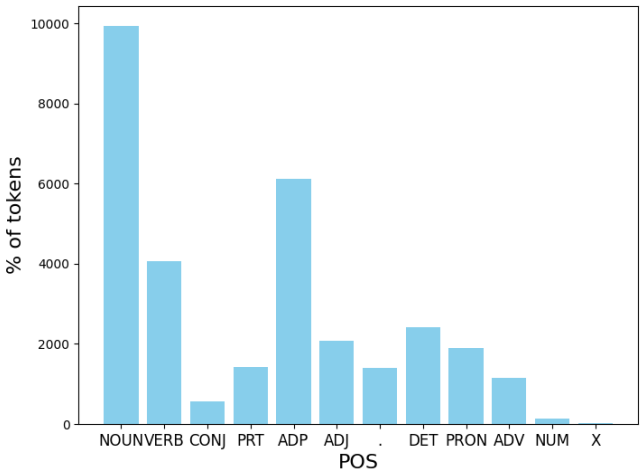


Figure 6. POS of samples that reach top-1 saturation in first 85% of layers of Llama3-8B

The pair is *discordant* if:

$$\begin{aligned}
&(x_i > x_j \text{ and } y_i < y_j) \quad \text{or} \\
&(x_i > x_j \text{ and } y_i > y_j) \quad \text{or} \\
&(x_i = x_j \text{ and } y_i \neq y_j) \quad \text{or} \\
&(x_i \neq x_j \text{ and } y_i = y_j)
\end{aligned}$$

The coefficient τ_{strict}, is computed as:

$$\tau_{\text{strict}} = \frac{C - D}{C + D},$$

where C is the number of concordant pairs, and D is the number of discordant pairs (including ties), ranging in values between $[-1, 1]$.

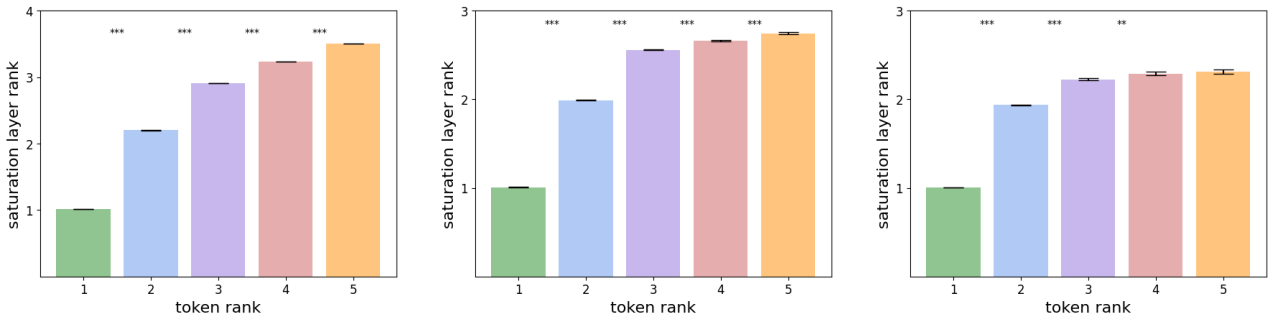
To check whether the sequence of saturation layers of the top-k tokens $(l^1, .., l^k)$ is strictly increasing, we use that sequence as one ranking, and the sequence $(1, 2, .., k)$ as the other. k is set independently for each token in the input to be the largest token index such that this token's reaches saturation by our definition i.e. $l^k < N$. Table 5 summarizes the results of this metric across the different models discussed in the paper. We find that for all models, the average τ coefficient indicates moderate agreement between the rankings, and is larger than all values over 1K permutations, where the saturation layers sequence were randomly shuffled for each instance, resulting in $p < 0.001$.

A.4. Reproducing Ordered Saturation in Various Text Transformers

We extend the results from Section 3.1 by reproducing the findings with additional decoder-only LLMs: GPT2-XL, and 8-bit quantized versions of Mistral-7B and Falcon-7B. Figure 7 presents the high correspondence between saturation layer and token rank observed consistently across all three models.

Using 1K randomly sampled questions from the MMLU dataset for each model and using the prompt format described in Section A.2, we find that the average rank of the k-th saturation layer increases monotonically with k (up to $k = 5$), with statistically significant differences between each consecutive token rank ($p < 0.001$, pairwise independent samples t-test). Similarly, we observe that Mistral's top-k tokens reach saturation in order of their ranking up to and including the 5th token, while Falcon's tokens follow this pattern up to the 4th ranking token.

Table 5 shows additional validation of this agreement between token ranking and saturation layer ranking using a stricter version of Kendall's τ metric as described in Section 3.1, with statistically significant average τ values for all models confirmed by a random permutation test.



(a) GPT2-XL (b) Mistral-7B (c) Falcon–7B

Figure 7. Average rank of the k-th saturation layer among the saturation layers for k=1,..,5 with standard error bars. Asterisks indicate statistically significant differences between consecutive token ranks (*** represents $p < 0.001$, ** represents $p < 0.01$), based on an independent samples t-test.

A.5. Reproducing Task Probing in Various Text Transformers

In support of our proposed task transition mechanism, using embeddings extracted during inference over 500 questions randomly sampled from MMLU dataset, we demonstrate that task index can be predicted from GPT2, Mistral and Falcon hidden layers' embeddings with high accuracy. We report full results and control settings in Table 6. Asterisks indicate statistically significant accuracy (*** represents $p < 0.001$), based on an Binomial Distribution probability test.

A.6. Training Data for Task Transition Classifier

Table 7 shows from which layers we took embeddings to train the task-transition classifier for each model.

A.7. Per Class Metrics for Task Transition Classifier

Table 8 shows accuracy scores per-class for each model, while Table 9 shows ROC-AUC scores per-class for each model.

A.8. Reproducing Ordered Saturation and Task Probing in LLaVa

We extend the results from Section 3.2 to multi-modal models by reproducing the findings with vision language model LLaVa-1.5-7B model (Liu et al., 2023).

Using 1K randomly sampled questions from the visual question answering MMMU dataset (Yue et al., 2024) and using the prompt format described in Section A.2, we find that the average rank of the k-th saturation layer increases monotonically with k (up to $k = 5$), with statistically significant differences between each consecutive token rank ($p < 0.001$, pairwise independent samples t-test). See Figure 8.

Table 5. Stricter Kendall's tau coefficients and p-values for each model. * signifies statistical significance with $p_{value} < 0.001$

Model	τ_{strict} **(avg $\pm$ ste)**
Llama3-8B (pre-trained)	$0.249^* \pm 0.004$
Llama3-8B (random initialization)	$0.283^* \pm 0.004$
ViT-L/16 (pre-trained)	$0.149^* \pm 0.007$
Whisper-large (pre-trained)	$0.210^* \pm 0.009$
GPT2-XL (pre-trained)	$0.187^* \pm 0.004$
Mistral-7B (pre-trained)	$0.312^* \pm 0.002$
Falcon-7B (pre-trained)	$0.234^* \pm 0.005$

Table 6. Accuracy of task number logistic regression classifier showing that in all modalities the layer embeddings contain information about the task number.

Model	**Accuracy**		
Layer embeddings	Random embeddings	Chance	
GPT2-XL (pretrained)	**84.2**$^{***} \pm 0.4$	20.4 ± 0.5	20.0
Mistral-7B (pre-trained)	**85.8**$^{***} \pm 0.1$	27.1 ± 0.1	25
Falcon-7B (pre-trained)	**91.0**$^{***} \pm 0.1$	24.6 ± 0.1	25
LLaVa-1.5-7B (pre-trained)	**89.0**$^{***} \pm 0.7$	25.7 ± 0.1	25
Qwen-Audio (pre-trained)	**95.1**$^{***} \pm 0.3$	25.0 ± 0.5	25

Table 7. Task transition probing data.

Model	**Layers**	**Dataset size**
Llama3-8B (pre-trained)	$19-27$	$8K$
Llama3-8B (random initalization)	$21-27$	$4K$
GPT2-XL (pre-trained)	$27-40$	$10K$
Mistral-7B (pre-trained)	$22-27$	$4K$
Falcon-7B (pre-trained)	$19-27$	$5K$
ViT-L/16 (pre-trained)	$16-21$	$2K$
Whisper-large (pre-trained)	$29-32$	$4K$
LLaVa-1.5-7B (pre-trained)	$22-27$	$4K$
Qwen-Audio (pre-trained)	$25-30$	$5K$

In support of our proposed task transition mechanism, using embeddings extracted during inference over 500 questions randomly sampled from MMMU dataset, we show in Table 6 that task index can be predicted from LLaVa's hidden layers' embeddings with high accuracy.

A.9. Reproducing Ordered Saturation and Task Probing in Qwen-Audio

We extend the results from Section 3.3 to another variant of Transformers with Qwen-Audio (Chu et al., 2023) a multimodal speech model trained for speech recognition, audio captioning, and audio-text retrieval.

Using 1K randomly sampled audios from the LibriSpeech we find that the average rank of the k-th saturation layer increases monotonically with k (up to $k = 4$), with statistically significant differences between each consecutive token rank ($p < 0.001$, pairwise independent samples t-test). See Figure 9.

In support of our proposed task transition mechanism, using embeddings extracted during inference over the same audios, we show in Table 6 that task index can be predicted from Qwen-Audio's hidden layers' embeddings with high accuracy.

A.10. Intervention Procedure Additional Details

Formally, this procedure consists of the following steps:

1. Given an input sequence $x =< x_1, ..., x_t >$ we first pass it through the model as in regular inference while storing the activation values at all hidden layers, i.e h_i^l for all $1 \leq i \leq t$, $1 \leq l \leq N$.
2. We calculate the saturation layer l_i^1 of the 1st token for each token w_i in the text.
3. We sample pairs of token indexes i, j in the text that the satisfy the following conditions:
   (a) The distance between i and j is no more than 40 tokens, i.e. $|i - j| \leq 40$.
   This is a precaution to minimize the effect of the difference in context on the model's predictions after intervention.

Table 8. Task transition probing per-class accuracy scores.

Model	Task 1	Task 2	Task 3	Task 4	Task 5
Llama3-8B (pre-trained)	0.82	0.89	0.93	–	–
Llama3-8B (random initalization)	0.69	0.79	0.89	–	–
ViT-L/16 (pre-trained)	0.65	0.62	0.60	0.64	0.68
Whisper-large (pre-trained)	0.52	0.53	0.48	0.58	–

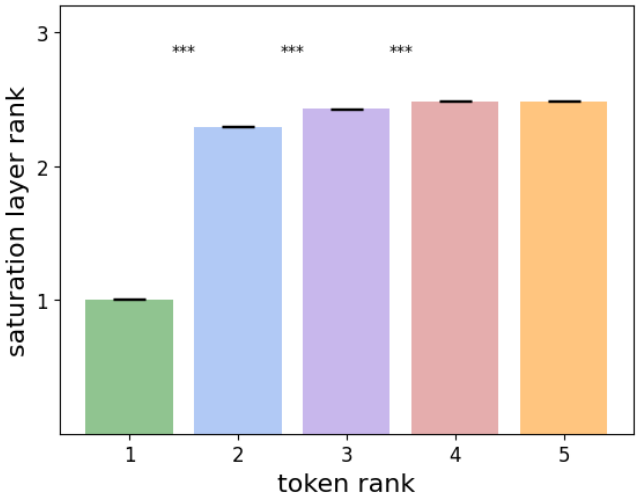


Figure 8. LLaVa-1.5-7B: Average rank of the k-th saturation layer among the saturation layers for k=1,..,5 with standard error bars. Asterisks indicate statistically significant differences between consecutive token ranks (*** represents $p < 0.001$, ** represents $p < 0.01$), based on an independent samples t-test.

(b) The model's top-1 prediction (in the final layer) for both indexes is the same token y, meaning $y = argmax(softmax(Eh_i^N)) = argmax(softmax(Eh_j^N))$.
The goal here is to avoid a conflict in the top-1 predictions which could be a confounding factor.

(c) There is a difference of at least 5 layers between the 1st token saturation layers of i and j, such that $|l_i^1 - l_j^1| \geq 5$, to ensure that the change in saturation layer after intervention is significant.

For convenience's sake we will assume in the remainder of the procedure description that $l_i^1 < l_j^1$, i.e that the saturation layer of the first index in the pair is smaller then that of the second index (even though both cases are allowed by our conditions).

4. We perform 7 additional forward passes, each time injecting the output from layer l' in range $[l_i^1 - 3, l_i^1 + 3]$ at position i as input into layer $l' + 1$ at position j . The goal here is to to quantify the difference in effect between layers preceding the saturation event and those after it.

Table 9. Task transition probing per-class ROC-AUC scores.

Model	Task 1	Task 2	Task 3	Task 4	Task 5
Llama3-8B (pre-trained)	0.94	0.97	0.97	–	–
Llama3-8B (random initialization)	0.87	0.90	0.96	–	–
ViT-L/16 (pre-trained)	0.86	0.82	0.81	0.83	0.87
Whisper-large (pre-trained)	0.78	0.77	0.69	0.77	–

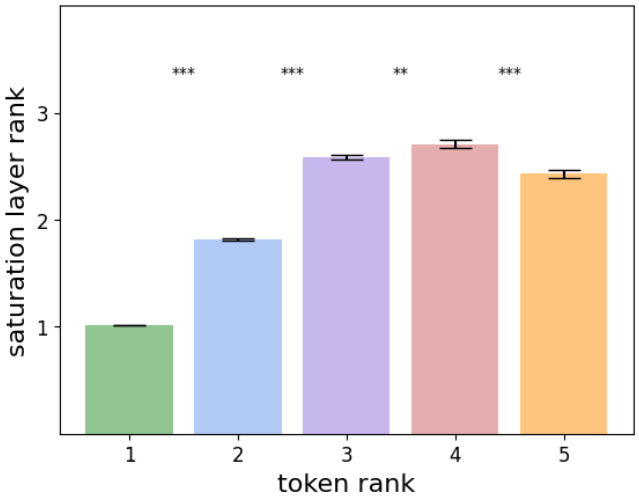


Figure 9. Qwen-Audio: Average rank of the k-th saturation layer among the saturation layers for k=1,..,5 with standard error bars. Asterisks indicate statistically significant differences between consecutive token ranks (*** represents $p < 0.001$, ** represents $p < 0.01$), based on an independent samples t-test.

5. We measure the causal effect of the intervention by calculating the percent of examples where the saturation layer of the 1st token after intervention $\tilde{l}_j^1$ is the layer on which we intervened, i.e. $\tilde{l}_j^1 = l + 1$.

For example, in the setting depicted in 4 we would take the indexes of the marked tokens "wanted" and "the" as our pair, where the original top-1 prediction in both is "artist". The top-1 saturation layer in the clean run for "wanted" is layer 20, so we would inject activations from layers 17 to 23 one at a time as inputs into the corresponding subsequent layers in the run of token "the" (24), and check for each one if the injected layer became the new top-1 saturation layer after the intervention.

A.10.1. Intervention Procedure on ViT

To adapt the intervention procedure described in section 4.2 to ViT-L/16 and the image classification setting we made the following modifications:

1. Since each image is processed independently by the model there is no need for two images to share a context, so the only requirements for two images to be chosen as a relevant pair were: a distance of at least 5 layers between the top-1 saturation layers, and the same top-1 class prediction in the final layer.

2. For each image, as in all experiments conducted on this model we only consider the prediction at index 0 corresponding to the [CLS] token in the input.

Figure 5 shows that the results for this model follow a similar step function pattern to the ones for Llama3-8B, where injecting embeddings from the top-1 saturation layer or one of the subsequent layers causes the model to "immediately" (at the injected layer) switch to the second task in a high percentage of cases, when compared to injecting embeddings from one of the layer before the top-1 saturation which almost never has the same effect.

A.10.2. Intervention Procedure on Whisper

We made the following adjustment to run the procedure described in section 4.2 on Whisper-large and 200 token pairs from randomly sampled 50 audios from the LibriSpeech dataset: Since the average audio in LibriSpeech is 10 seconds long there are not enough tokens in one sample to find relevant pairs, so we wave the requirement for a pair to share context and only leave two conditions: a distance of at least 5 layers between the top-1 saturation layers, and the same top-1 prediction in the final layer.

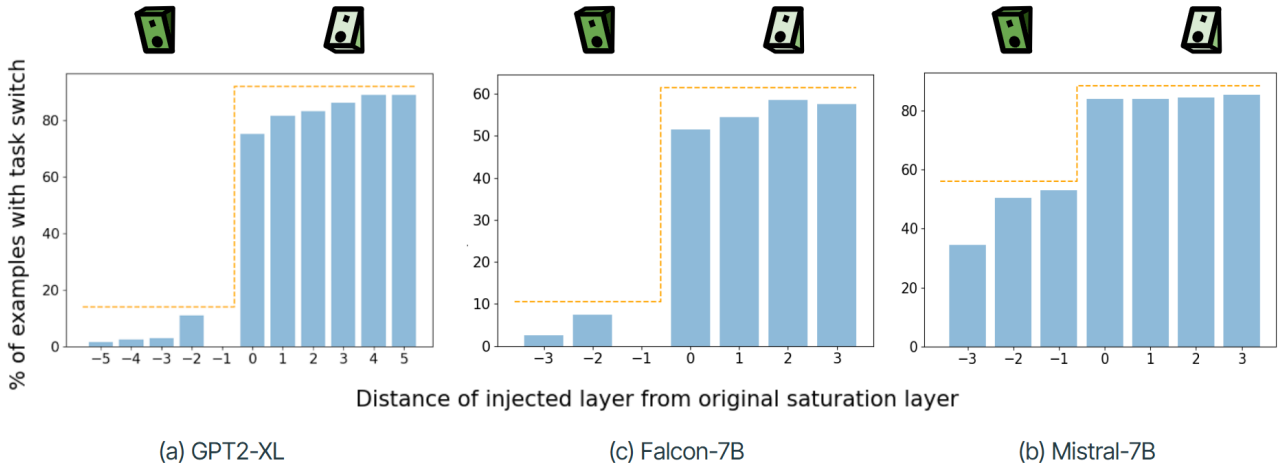


Figure 10. Flipping the Top-1 Switch. The percentage of examples where the top-1 saturation occurred at the injected layer after the intervention, shown as a function of the layer from which the injected activations were taken, relative to the original saturation layer (e.g., -2 means activations were taken from two layers before the original saturation layer).

Figure 5 shows that the results for this model follow a similar pattern to the other two models, even though the effect increases in the following layers after the saturation event.

A.11. Reproducing Task Switch in Various Text Transformers

We show in Figure 10 the results of the intervention procedure described in Section 4.2 using GPT2-XL, Mistral-7B and Falcon-7B models over 200 token pairs (each) extracted from 10 randomly sampled texts from CNN/DM dataset. As with the other models, when the injected activations are taken from the top-1 saturation layer or later layers, the new top-1 saturation happens at the injected layer much more frequently than when injecting activations from earlier layers, indicating that these layer contain the signal to switch to the next task and keep the top-1 constant.

A.12. Additional Details for Token-Level Early Exit Measures Comparison

Table 10 shows the accuracy and speedup ratio of the Softmax Response token-level early-exit strategy at various thresholds.

Table 11 shows the accuracy and speedup ratio of the Hidden-state saturation token-level early-exit strategy at various thresholds.

Figure 11 visualizes the performance-efficiency trade-off of both methods in comparison to our novel early-exit strategy, as well as baseline and oracle.

Table 10. Softmax Response accuracy & speedup ratios at various confidence thresholds

	Thresholds								
	0.5	0.6	0.7	0.8	0.9	0.92	0.94	0.96	0.98
Accuracy	33.149	34.861	36.437	37.930	39.334	39.717	40.026	40.340	40.648
Speedup Ratio	1.326	1.284	1.243	1.199	1.145	1.132	1.117	1.099	1.073

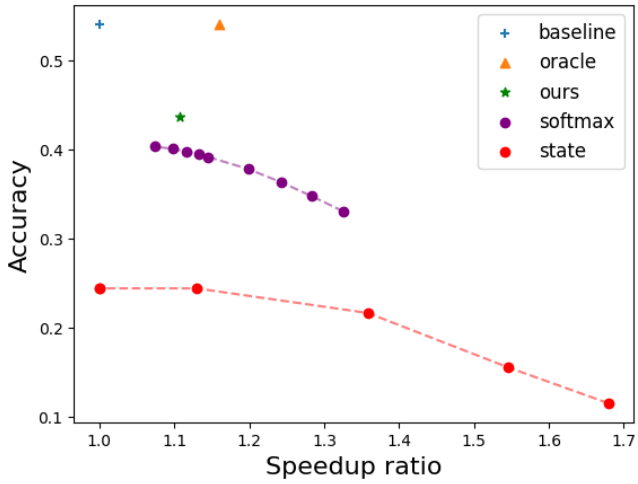


Figure 11. Performance-efficiency trade-off comparison of different confidence measures against a static baseline (where all layers are used for each token) and a local oracle measure (where the early exit is at the top-1 saturation layer). The graph shows softmax and state confidence measure results at different thresholds. Our method achieves the highest next-word prediction accuracy out of all early-exist methods while providing significant speedup compared to the baseline.

Table 11. Hidden-sate saturation accuracy & speedup ratios at various confidence thresholds

	Thresholds					
	0.9	0.92	0.94	0.96	0.98	0.99
Accuracy	11.267	15.183	21.396	24.250	24.252	24.252
Speedup Ratio	1.682	1.552	1.361	1.130	1.001	1.001